

## Practice Question -Longest Increasing Subsequence

```
package arthematicoperations;
import java.util.Arrays;
public class Longestsubsequent {
    public static void main(String[] args) {
        int[] arr = {3, 10, 2, 1, 20};
        int n = arr.length;
        int[] lis = new int[n];
        // Initialize LIS values for all indices
        Arrays.fill(lis, 1);
        // Compute optimized LIS values in bottom up manner
        for (int i = 1; i < n; i++) {
            for (int j = 0; j < i; j++) {
                if (arr[i] > arr[j] && lis[i] < lis[j] + 1) {
                    lis[i] = lis[j] + 1;
                }
            }
        }
        // Pick maximum of all LIS values
        int maxLis = Integer.MIN_VALUE;
        for (int i = 0; i < n; i++) {
            if (lis[i] > maxLis) {
                maxLis = lis[i];
            }
        }
        System.out.println("Length of longest increasing subsequence is: " +
maxLis);
    }
}
```

The screenshot shows the Eclipse IDE with a Java file named `Longestsubsequent.java`. The code implements a dynamic programming solution for the Longest Increasing Subsequence problem. It uses a bottom-up approach, initializing an array `lis` with 1s and then iteratively updating it based on the values of the input array `arr`. The final result is the maximum value in the `lis` array.

```
1 package arthematicoperations;
2
3 import java.util.Arrays;
4
5 public class Longestsubsequent {
6     public static void main(String[] args) {
7         int[] arr = {3, 10, 2, 1, 20};
8         int n = arr.length;
9         int[] lis = new int[n];
10
11         // Initialize LIS values for all indices
12         Arrays.fill(lis, 1);
13
14         // Compute optimized LIS values in bottom up manner
15         for (int i = 1; i < n; i++) {
16             for (int j = 0; j < i; j++) {
17                 if (arr[i] > arr[j] && lis[i] < lis[j] + 1) {
18                     lis[i] = lis[j] + 1;
19                 }
20             }
21         }
22
23         // Pick maximum of all LIS values
24         int maxLis = Integer.MIN_VALUE;
25         for (int i = 0; i < n; i++) {
26             if (lis[i] > maxLis) {
27                 maxLis = lis[i];
28             }
29         }
30         System.out.println("Length of longest increasing subsequence is: " + maxLis);
31     }
32 }
```

The console output shows the program terminated successfully and printed the result:

```
<terminated> Longestsubsequent [Java Application] C:\Users\Swaroop\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (15-N
Length of longest increasing subsequence is: 3
```

Git commands

Git add Practice Question-Longest Increasing Subsequence.pdf

Git commit -m "Changes made"

Git push -u origin master